IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1-7. (Cancelled)

8. (Previously Presented) A system for distributed filtering of samples within a convolution kernel to calculate values for a corresponding pixel comprising:

a series of means for calculating partial sums, wherein each member of the series calculates partial sums for a corresponding portion of the samples within the convolution kernel for the pixel, wherein the partial sums calculated by each member of the series comprise 1) a sum of weights determined for the sample locations in the corresponding portion of samples and 2) a sum of weighted sample values for the corresponding portion of samples, wherein each member of the series adds the calculated partial sums to corresponding accumulated partial sums and outputs the new accumulated partial sums, and wherein a last member of the series calculates pixel values from the final accumulated partial sums; and

means for passing accumulated partial sums from one member of the series to the next;

a plurality of means for storing sample values, wherein each means for storing sample values is dedicated to a different member of the series; and

a plurality of means for rendering samples, wherein each sample generated is stored in one of the means for storing sample values.

9. (Original) The system of claim 8, wherein a sample comprises parameter values for color and transparency, and wherein partial sums comprise partial sums for each of the parameter values.

10. (Original) The system of claim 8, wherein each means for calculating partial sums is assigned one or more sample bins from an interleaved array of sample bins and the interleaved array of sample bins is repeated across screen space.

11. - 12. (Cancelled).

13. (Previously Presented) The system of claim 8, further comprising a means for converting the pixel values to video output signals.

14. (Cancelled)

15. (Currently Amended) The system of claim [[14]]20, further comprising a memory (k) dedicated to sample manager (k), wherein memory (k) stores sample data for samples in the sub-set of screen space assigned to the sample manager (k).

16. (Original) The system of claim 15, wherein each memory (k) comprises a plurality of memory units.

17. (Original) The system of claim 15, wherein sample manager (k) is operable to read the set of samples from the memory (k).

18-19. (Cancelled).

20. (Currently Amended) ~~The system of claim 14,~~A system for distributed filtering of samples comprising:

a series of N sample managers (k), wherein k is an integer with range 0 to N-1, and wherein N is an integer greater than 1;

a partial sums bus connecting each sample manager (k) in the series of sample managers to the next sample manager (k+1);

wherein sample manager (k) is operable to:

receive accumulated partial sums from a prior sample manager (k-1), if k is greater than zero, wherein each sample comprises values for a plurality of parameters, and wherein partial sums comprise partial sums for each sample parameter value,

calculate partial sums for a set of samples, wherein the set of samples are within a sub-set of screen space assigned to sample manager (k), and wherein the set of samples are located within a convolution kernel defined for a pixel,

add the partial sums to the accumulated partial sums, and

output the accumulated partial sums to sample manager (k+1), if k is less than N-1; and

wherein a designated sample manager is operable to calculate pixel parameter values from the final accumulated partial sums; and

wherein each pixel parameter value equals a corresponding final accumulated sum of weighted sample parameter values for each sample within the convolution kernel divided by an accumulated sum of weights for locations of each sample within the convolution kernel.

21-39. (Cancelled).

40. (Currently Amended) A method for distributed filtering of samples comprising:

calculating first partial sums in a first filter unit for a first set of samples, wherein partial sums comprise a sum of weighted sample values for the set of samples and a sum of [[the]] weights determined for the locations of each sample, wherein a weighted sample value is a product of the sample value and the determined weight for the location of the sample, wherein sample values comprise color values and transparency, wherein the first set of samples is a portion of the samples located within a convolution kernel defined for a pixel location, and wherein the first set of samples are within a region of screen space assigned to the first filter unit; and

sending the first partial sums to a sequence of additional filter units, wherein each of the additional filter units:

receives accumulated partial sums from the previous filter unit,

calculates new partial sums for a corresponding set of samples located within the convolution kernel and within a corresponding region of screen space assigned to the filter unit,

adds the new partial sums to the accumulated partial sums, and

if not a last filter unit in the sequence of filter units, sends the new accumulated partial sums to the next filter unit in the sequence of filter units~~; and~~

~~wherein a last filter unit in the sequence of filter units:~~

~~receives accumulated partial sums from the previous filter unit in the sequence,~~

~~calculates new partial sums for a corresponding portion of samples, and~~

~~adds the new partial sums to the accumulated partial sums to complete final accumulated partial sums~~.

41. (Currently Amended) A method for distributed filtering of samples comprising:

calculating first partial sums in a first filter unit for a first set of samples, wherein the first set of samples is a portion of the samples located within a convolution kernel defined for a pixel location, and wherein the first set of samples are within a region of screen space assigned to the first filter unit; and

sending the first partial sums to a sequence of additional filter units, wherein each of the additional filter units:

receives accumulated partial sums from the previous filter unit,

calculates new partial sums for a corresponding set of samples located within the convolution kernel and within a corresponding region of screen space assigned to the filter unit,

adds the new partial sums to the accumulated partial sums, and

if not a last filter unit in the sequence of filter units, sends the new accumulated partial sums to the next filter unit in the sequence of filter units~~; and~~

~~wherein a last filter unit in the sequence of filter units:~~

~~receives accumulated partial sums from the previous filter unit in the sequence,~~

~~calculates new partial sums for a corresponding portion of samples,~~

~~calculates final sums by adding the new partial sums to the accumulated partial sums~~, and

if the last filter unit in the sequence of filter units, calculates pixel values from the final sums, by multiplying a final sum of weighted sample values times a reciprocal of a final sum of the weights for each parameter value comprising one or more of color values and transparency.

42-43. (Cancelled)

44. (New) The system of claim 8, wherein the last member of the series calculates pixel values from the final accumulated partial sums by dividing a final accumulated sum of weighted sample values by a final accumulated sum of weights.

45. (New) The system of claim 8, wherein for each member of the series the corresponding portion of samples resides in a sub-set of screen space and the sub-set is interleaved across screen space.

46. (New) The system of claim 45, wherein the members of the series comprise 16 means for calculating partial sums, and wherein each means for calculating partial sums addresses one sample bin in a 4 by 4 array of sample bins and a corresponding sample bin in each repetition of the 4 by 4 array to span screen space.

47. (New) The system of claim 8, wherein the members of the series are arranged in a plurality of groups, and wherein each group's means for calculating partial sums are connected in series.

48. (New) The system of claim 47, wherein for each means for calculating partial sums the corresponding portion of samples resides in a corresponding sub-set of screen space and the sub-sets are interleaved across screen space, wherein for a system of 4 groups of 4 means for calculating partial sums in a series, each means for calculating partial sums within a group addresses one sample bin in a 2 by 2 array of sample bins that is repeated across a 16 by 16 array of sample bins, and wherein four permutations of each of the four different 16 by 16 arrays (one for each group) are combined to form a 64 by 64 array of sample bins that is repeated across screen space.

49. (New) The system of claim 47, wherein a designated one of the means for calculating partial sums in each group calculates pixel values from the final accumulated partial sums.

50. (New) The method of claim 40, wherein the convolution kernel defined for the pixel is a region in screen space within a defined boundary and centered on the pixel location in screen space.

51. (New) The method of claim 40, wherein a weight for the location of each sample value is determined by a weight function selected from a set of functions comprising a box filter, pyramid filter, circular filter, cone filter, Gaussian filter, and sinc filter.

52. (New) The method of claim 40, wherein the last filter unit of the sequence of filter units calculates pixel values from the final accumulated sums.

53. (New) The method of claim 40, wherein each set of samples is within a screen space region assigned to the corresponding filter unit, and wherein each set of samples are read from a memory dedicated to the corresponding filter unit.

54. (New) The method of claim 40, wherein for each filter unit, the corresponding set of samples resides in a sub-set of screen space and the sub-sets are interleaved across screen space.

55. (New) The method of claim 54, wherein for a system of 16 filter units, each filter unit addresses one sample bin in a 4 by 4 array of sample bins that is repeated across screen space.

56. (New) The method of claim 40, wherein the filter units are sub-divided into a plurality of groups of filter units and a plurality of partial sums buses interconnect the groups, and wherein each group is a series of filter units.

57. (New) The method of claim 56, wherein a last filter unit in each group calculates values for a pixel from the final accumulated partial sums corresponding to the pixel.

58. (New) The method of claim 56, wherein for each filter unit the corresponding set of samples resides in a sub-set of screen space and the sub-sets are interleaved across screen space.

59. (New) The method of claim 58, wherein for a system of 4 groups of 4 filter units in a series, each filter unit within a group addresses a sample bin in a 2 by 2 array of sample bins that is repeated across a 16 by 16 array, and wherein four permutations of each of the four different 16 by 16 arrays (one for each group) are combined to form a 64 by 64 array of sample bins that is repeated across screen space.